Instruction of compilation and usage for

Simple Ontology Visualization API

Table of contents:

1. Description of project

Project SOVA, created in Maven, consists from 3 modules:

- library SOVA (in folder: SOVA),
- plugin to Protege application (in folder: pluginProtegeSova),
- demo – standalone application, which can simulate functions of SOVA library without installing plugin in Protege application (in folder: demo).

Project created in MAVEN allows to build all modules at once or singly.

1.1 Description of SOVA library module

SOVA library module is main part of project, which is responsible for visualization.

1.2 Description of module, which is plugin to Protege application

Module pluginProtegeSova creates file with .jar format, which allows visualization graphs in Protege application. To be able to use it in program Protege, this file supposed to be place in folder `plugins` of this program.

1.3 Description of demo module

Module demo allows to demonstrate functionalities of SOVA library without installing plugin in Protege program. Module creates file with .jar format, which is standalone application.

1.3.1 Preparing demo module to building

Module demo have to be prepared before usage.
It is necessary to find file Constants.java, which located in `root of project\demo\src\main\java\org\pg\eti\kask\sova\demo`. This file supposed to be open with any text editor.

Find line, which contains:
`public static final String ONTO_TEST_DIRECTORY = "X:\\Repozytoria\\sova\\doc\\OWL\\onot-test.owl";`
Declared value in this line is local path to ontology, which will be visualized in demo application. This value should be set individually for every user before building this module. Examples of visualizations can be found in localization `root of project\doc\OWL`.

Another modification supposed to be made in line, which contains:
`public static final String PROPERTIES = "X:\\Repozytoria\\sova\\SOVA\\src\\visualization.properties";`
Declared value in this line is path to file, which contains properties of visualization. This value supposed to contain path to location where is place `visualization.properties` file. This value should be set individually for every user before building this module. This file can be found in `root of project\SOVA\src\visualization.properties`.

Demo module is ready to build.

1.3.2 Launching standalone demo application

To execute demo application, it is necessary to open command line in localization `root of project\demo\target` and call following command:

```
java -jar DEMO-0.8.5-jar-with-dependencies.jar
```

2. Building project

2.1 Installation of necessary libraries

Before first building project, library owlapi-bin have to be installed, because it is needed to built modules. To install it, it is necessary to open command line in folder lib, which is in main catalog of project (root) and call following command:

```
mvn install:install-file -Dfile=owlapi-bin.jar -DgroupId=owlapi-bin -DartifactId=owlapi-bin -Dversion=1 -Dpackaging=jar
```

This command will install owlapi-bin library in MAVEN's local repository and modules will be able to use them during building.

2.2 Building all modules at once

To build project, it is necessary to open command line in main catalog of project (root) and call following command:

```
mvn install
```

All modules will be build at once in following order:

- SOVA,
- pluginProtegeSova,
- demo.

File with .jar format will be create for each module in localization `root of project/[name of module]/target` (modules with name of folder is described at point 1.).

2.3 Building modules singly

Each module can be built singly. To build chosen module, open command line in folder, which contains it (modules with name of folder is described at point 1.) and call following command:

```
mvn install
```

File with .jar format will be create in localization `root of project/[name of module]/target`

Module demo and pluginProtegeSova can't be built without building SOVA module at least once before. Library SOVA will be placed in MAVEN's local repository and other modules will be using it during building.

3. Changing version number of project

To change version number of project, you have to make following modifications. At the beginning, open file pom.xml, which is placed in main catalog of project (root) and find line, which contains tag <version> and </version>. Value, which is placed between this tags is version number of project.

In the next step, you need to open pom.xml file, which is placed in module SOVA (`root of project/SOVA/pom.xml`). To modify this file, find part which contains following lines:

```
<parent>
	<groupId>org.pg.eti.kask.ont.pluginSova</groupId>
	<artifactId>pluginSovaBuild</artifactId>
	<version>X.X.X</version>
	<relativePath>../pom.xml</relativePath>
</parent>
```

In this part, you need to change value between <version> tags. Value have to be the same as value, which was set in pom.xml in main catalog of project at the beginning.

In module demo and pluginProtegeSova, you need to repeat steps, which was made to modify module SOVA.